

REMARKS

In response to the Office Action mailed May 19, 2005, Applicants respectfully request reconsideration. Claims 1, 3-15, 17-51, 53-64 and 66-95 were previously pending in this application. By this amendment, Applicants cancel all previously-pending claims without prejudice are disclaimer. Applicants expressly reserve the right to continue to pursue these claims in one more continuing applications. Claims 96-113 are newly added in this application. As a result of these amendments, claims 96-113 are currently pending and the application as presented is believed to be in condition for allowance.

Initially, Applicants thank Examiner Vu for the courtesies extended during the telephone interview conducted on November 14, 2005, the substance of which is summarized herein.

During the telephone interview, Applicants indicated that it appeared none of the references applied in the rejection (i.e., Richburg, Kim, Underwood, or Rudolph) disclose using a design matrix to automatically generate code for a software system. Applicants requested that the Examiner explain the portions of the references there were being relied on as purportedly disclosing this aspect. While the Examiner did not cite specific portions of the references, the Examiner indicated that he believed the use of matrices in the design of software was known prior to Applicants' invention.

Applicants maintain that none of the references applied in the rejection of the claims as previously-pending discloses the automatic generation of software code from a design matrix. However, to further the prosecution of this application, Applicants have canceled the previously-pending claims added new claims 96-113 which relate generally to how a design matrix may be created and used to define a software system.

Specifically, claim 96 is directed to a method of designing a software system. The method comprises: defining a set of functional requirements that describe what the software system is to achieve; defining a set of design parameters, where each design parameter in the set satisfies at least one of the functional requirements; decomposing the set of functional requirements and design parameters to create a hierarchy of functional requirements and a hierarchy of design parameters, wherein at least one functional requirement of the set of functional requirements is a parent functional requirement at a first level in the hierarchy of functional requirements and is decomposed into at least two child functional requirements at a

second level in the hierarchy that is below the first level, and wherein the at least two child functional requirements collectively accomplish the parent functional requirement; defining a design matrix that maps each design parameter in the hierarchy of design parameters to the at least one functional requirement in the hierarchy of functional requirements that the respective design parameter satisfies; and using the design matrix to define an object-oriented structure of the software system, wherein at least one functional requirement in the hierarchy of functional requirements represents a software object of the software system, and wherein at least one design parameter in the hierarchy of design parameters represents an input to the software object.

None of the references cited in the Office Action (i.e., Richburg, Kim, Underwood, or Rudolph) discloses or suggests, “defining a design matrix that maps each design parameter in the hierarchy of design parameters to the at least one functional requirement in the hierarchy of functional requirements that the respective design parameter satisfies” and “using the design matrix to define an object-oriented structure of the software system, wherein at least one functional requirement in the hierarchy of functional requirements represents a software object of the software system, and wherein at least one design parameter in the hierarchy of design parameters represents an input to the software object,” as recited in claim 96.

While Kim discloses a design matrix, Kim does not disclose or suggest that the design matrix may be used to define an object-oriented structure of a software system. Nowhere does Kim disclose that a functional requirement in the design matrix represents a software object and a design parameter represents an input to the software object. Similarly, Rudolph discloses a design matrix that maps between the functional requirements and the design parameters of a design, but does not even disclose that the design matrix may be used in the design software, let alone define an object-oriented structure for a software system. Rather, Rudolph only discloses that a design matrix is used in the design of mechanical systems.

Richburg is directed to the generation of program code files using application database files and knowledge base files (column 5, lines 7-9). The knowledge base files include program code in the most general form for solution of a problem (column 5, lines 45-47). The application database files include the requirements of the specific application. A program processor uses the instructions from the knowledge base file along with information in the application database files to generate an output file (column 6, lines 10-14). The output file includes code from the

knowledge base files customized to a particular application as specified by the application database files (column 6, lines 14-22). Nowhere does Richburg disclose or suggest that an object-oriented structure of a software system may be designed using a design matrix. Indeed, Richburg is completely unrelated to the design of a software system using a design matrix.

Underwood is directed to interfacing servers in a Java based e-commerce architecture. The Abstract of Underwood discloses a first server and second server with a proxy component situated therebetween. A request for a business object is identified by an application on the first server. The first server is then connected to the second server. Next, a selection criteria from the first server is transmitted to the second server. In response to the selection criteria, the first server receives a first recordset and a second recordset from the second server. Business data is included in the first recordset and result codes are included in the second recordset. The first and second recordsets are mapped to the business object and the business object is sent to the application on the first server. Like Richburg, Underwood is not related to the design of a software system using a design matrix. Underwood is not even related to software design, but rather is directed to transferring information over the World Wide Web.

In view of the foregoing, claim 96 patentably distinguishes over each reference cited in the Office Action. Claims 97-104 depend from claim 96 and are patentable for at least the same reasons.

Claim 105 is directed to at least one computer readable medium encoded with instructions that, when executed on a computer system, perform a method of allowing a user to define a software system. The method comprises: allowing the user to defined a set of functional requirements that describe what the software system is to achieve; allowing the user to define a set of design parameters, where each design parameter in the set satisfies at least one of the functional requirements; allowing the user to decompose the set of functional requirements and design parameters to create a hierarchy of functional requirements and a hierarchy of design parameters, wherein at least one functional requirement of the set of functional requirements is a parent functional requirement at a first level in the hierarchy of functional requirements and is capable of being decomposed into at least two child functional requirements at a second level in the hierarchy that is below the first level, and wherein the at least two child functional requirements collectively accomplish the parent functional requirement; allowing the user to

define a design matrix that maps each design parameter in the hierarchy of design parameters to the at least one functional requirement in the hierarchy of functional requirements that the respective design parameter satisfies; and using the design matrix to define an object-oriented structure of the software system, wherein at least one functional requirement in the hierarchy of functional requirements represents a software object of the software system, and wherein at least one design parameter in the hierarchy of design parameters represents an input to the software object.

As should be clear from the discussion above, none of the references discloses or suggests, “allowing the user to define a design matrix that maps each design parameter in the hierarchy of design parameters to the at least one functional requirement in the hierarchy of functional requirements that the respective design parameter satisfies” and “using the design matrix to define an object-oriented structure of the software system, wherein at least one functional requirement in the hierarchy of functional requirements represents a software object of the software system, and wherein at least one design parameter in the hierarchy of design parameters represents an input to the software object,” as recited in claim 105.

Thus, claim 105 patentably distinguishes over each of the references cited in the Office Action. Claims 106-113 depend from claim 105 and are patentable for at least the same reasons.

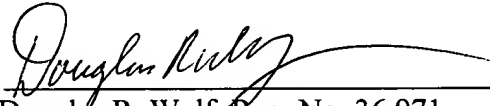
CONCLUSION

In view of the foregoing amendments and remarks, this application should now be in condition for allowance. A notice to this effect is respectfully requested. If the Examiner believes, after this amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully Submitted,

Sung-Hee Do et al., Applicant

By: 
Douglas R. Wolf, Reg. No. 36,971
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210-2211
Telephone: (617) 720-3500
Attorneys for Applicants

Docket No.: A0734.70001US00
Date: November 21, 2005
x11/19/05x